Department of Computer Science & Engineering
**Indian Institute of Information Technology Senapati**

# Property Management System with multiple layers using Django

(Course Code: CS 200)

Jay Dev Jha
18010117

Supervised by
Dr.Navanath Saharia

24 June 2020

**Abstract**

The project dealt with managing different layers of property on the basis of different parameters they acquire and can support with the facilities they accomplish, in order to ease with the searching and booking operations. The project is implemented using django due to its various incorporated features that includes a gist describing the creation of applications from underlying concepts to implementation without taking much time, along with the dominant characteristics of reassuringly secure, exceedingly scalable, fully loaded, incredibly versatile and many of such supporting aspects it integrates. The system has been designed to maintain the classic three layered architecture, that helps to keep the consistency among the flow sequence of the data within different layers. The developed system has also focused on the authentication and authorization on the numerous tasks involved by differenet layered parties. The users will have their own dashboards to keep a check on various properties they have booked and be able to track the operations performed so far. The overall sytem is so developed to enhance the simplicity while managing the properties from both the prespectives i.e., from admin and maintainers point of view as well as from the real world users.

# Contents

# List of Figures

# Chapter 1

# Introduction

The property management task may be considered as one of the difficult jobs when there is a long database of properties with significant differences in the features and charateristics they acquire which differentiates them among themselves. The system has been developed by building up the database for diverse properties Indian Insitute of Information Technlogy, Senapti acquires which constitutes classrooms, laboratories, auditorium hall, conference hall, admins and many such infrastructures classifying each other according to the departments and the requirements. Implementation using django helped me a lot to manage many different tasks confronted while building up the system. Django gives complete assistance in a way from conceptualizid building to the security issues. The upcoming chapters will help the readers to have an underlying view of the fundamental requirements implemented using the basic ER diagrams and the data flow sequence so as to get a clear picture of the processes incorported and their working principles.
The various subsections included in this chapter introduces with the issues confronted while managing the properties, maintaining the consistency among the different layered parties supporting the three layered architecture. The problem statement elucidates with the problems with the existing systems and the need of integrating and incorporating the required features and applications. The motivation to select the project and the tool to implement the same in described in the motivation section. The roadmap and the implementation plan elaborates the planning and the system software analysis necessary to achieve the desired the ideal objective.
The subsequent chapters describes the existing system study elaboratively and more concisely along with the necessary requirements. System analysis is so important as it helps to design systems where subsystems may have conflicting objectives. Other influencing factors to study them are helping to achieve inter compatibility and unity of the sub systems alon with understanding of complex structures. Above all, System analysis gives an advantage of understanding and comparing the subsystems functions with complete system. The comparative analysis demonstrates various missing features in the existing systems and which of those applications needed to be integrated so as to enhance the betterment of the system along with ease and simplicity. And also we have to consider the transfer of the large amount of data through the different layered parties will give errors while transferring. Nevertheless, sensitive data transfer is to be carried out even if there is lack of an alternative. Data security that includes authorization and authentication on it, in the existing system is the motivation factor for a new system with higher-level security standards for the information exchange.
The system design and architecture chapter narrates the base for building and developing the system. The implementation plan will take the readers to the actual development plan and the various experimental set-ups needed to develop the system. The corresponding chapters to this section of the proposed documentation will going to analyze the generated output and the various results to

be interpreted. The documentation focuses on the result assessment topic which gives an overview on the other side of the built system, i.e., the issues that may raise due to some of the invalid operations if so committed and the conflicting datas while uploading the same as the maintainer or the admin staffs.
The documentation concludes with the future scope of the project that further increase the searching and booking opeartion on the large datasets. It talks about integrating other relevant features of including the road map or the routes simplifying the approaching facility and giving the rough idea on how long it would take to reach the desired place along with the traffic details. The last section contains the list of references and bibliographies used to build and develop the system comprising of the referenced books, sites, papers and articles.

## 1.1 Problem statement

**Property Management** is one of the difficult tasks when there is a large sphere of properties classified based on the minute significant differences in terms of the fatures they incorporate. The present property management systems discusses and maintains the properties of the same characteristics such as hotel management, rental houses and cars and many more. they all narrates the story on the same base of the single entity or asset for instance, hotel. With the central aim of developing a more adaptable and accessible property management system, the chief target is to give a helping-hand to the users in booking the properties and assets according to their needs and requirements as necessary. The basic purpose is to maintain the hierarchy of the property management tasks between Superuser, Admin Staff and the real world users so as to increase feasibility, accessibility and adaptability.
Some of the systems does not maintain the basic three layer architecture and hence facing problems in data flow sequence and its consequences. The subsequent challenges then lies in the authentication and authorization fronts where there are high chances of data loss due to low maintenance of the data security. Again, to develop and build followed by the maintaining operation requires the precise tools and techniques that can support various applications with minimal time and smaller codes in order to debug easily and quickly. The implementation tool must go along with the data security and consistency among its flow.
The proposed system focuses on the following to eradicate some of the problems in the existing system:

- The login-logout mechanism must be properly managed in order to keep check on authentication and authorization process.
- The system design architecture should be followed according to the classic definition of three layered architecture so as to maintain the consistency among the data flow between different layered members of the parties.
- A separate framework and platform for both the real world users and the admin and the maintainer staffs subsequently, to be a more conventional and friendly system to operate.
- An easy to maintain dashboards for the actual users such that they can track their operations on different properties available to them.
- A conventional framework for the admin staffs to carry out various data management system operations with improved simplicity and well proofed tasks.

## 1.2 Motivation behind selection of the project

Database Management System is one of the core of Computer Science Engineering and working on it will prove to be very useful and thats the main reason behind motivation for selecting the project. This project involves the working of both the frontend as well as the backend portion of the system leading to learning on both the aspects of the management system. This project helped me a lot to understand the working of the database and how to keep a balance between the data flow among the various layered members of the proposed systems. With the central thought process that this project will guide me on how to deal with managing the tasks among the three parties i.e., superuser, admin staffs and the real world users has proved me itself after working on the same.

### 1.2.1 Motivation behind selecting the tool

As Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design, it takes care of much of the hassle of Web development, so that we can focus on writing our own app without needing to reinvent the wheel.
The three important features are:

1. Ridiculously fast implying that Django helps to make applications from concept to completion as quickly as possible.

2. Reassuringly secure i.e., Django takes security seriously and helps developers avoid many common security mistakes

3. Exceedingly scalable means that some of the busiest sites on the Web leverage Djangos ability to quickly and flexibly scale.

Other advantages includes:

- Fully loaded: Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks right out of the box.

- Incredibly versatile: Companies, organizations and governments have used Django to build all sorts of things from content management systems to social networks to scientific computing platforms.

## 1.3 Roadmap


Figure 1.1: A complete Roadmap

The above roadmap illustrates the features and the tasks each one of the including members of the system performs in order to have smooth functioning of the overall system. The superuser has the authority to grant access to its admin staffs who will look after the various languages relevant to database management system such as *Data Definition Language* (DDL), *Data Manipulation Language* (DML) and *Data Control Language* (DCL) along with the security and overall maintenance. The included properties in the database must have the following characteristics:

1. Name
2. Property Id
3. Location
4. Description
5. Establishment Data

These are followed by the accessibility provided to the real world users in terms of *view, filter search, vacancy info, booking slots, enquiring the desired information related to a particular asset, confirming it and finally reserving the proerty.* This roadmap helped me to get a clear idea about the objectives of various constituting members of the system.

## 1.4 Contribution

My contribution with respect to the proposed and designed system lies in developing and building up the complete ***Property Management System*** following the mentioned roadmap and focusing on the ideal aims and objectives. More specific to the system design, I have analyzed the system requirements using ER diagrams and UMLs followed by the implementation plan and testing. I have contributed from the underlying concept building to the final implementation with regular assistance from my supervison on fixing any bug and how to enhance the quality of the system. My contribution lies in the fact that, the first and the foremost aspect of understanding the end goals. The working mechanism flows sequentially from getting the thorough knowledge on the various interdependencies required for building up the system. The implementation plan is completed by breaking down the complete development process int various subtasks and setting the milestones and timeline for each.
The fundamental base of my contribution is to implement the conevntional framework maintaing the dependencies required for proper functioning. To put it in a nutshell, my tasks comprised the development of the whole system followed by the testing and fixing any confronted bugs, moving to the conclusion of writing the documentation and to tranlate them in an easy-to-get language for user interfaces. I have been dedicated to this project from the allocation stage to its final implementation going through the concept building and system design. My weekly progress report can be verfied through the upcoming gantt chart describing the various set goals and its corresponding milestones.

## 1.5 System/Software used

The whole project is implemented on Linux operating system. The software tool used in the development of the system is a high-level python web framework: ***Django***. Along with this tool, the project includes the basic CSS and HTML to give the frontend outlook. The database is maintained using SQLite for the backend portion of the project.

## 1.6 Implementation plan

The weekly progress report has been designed and planned using the following gantt chart [1]

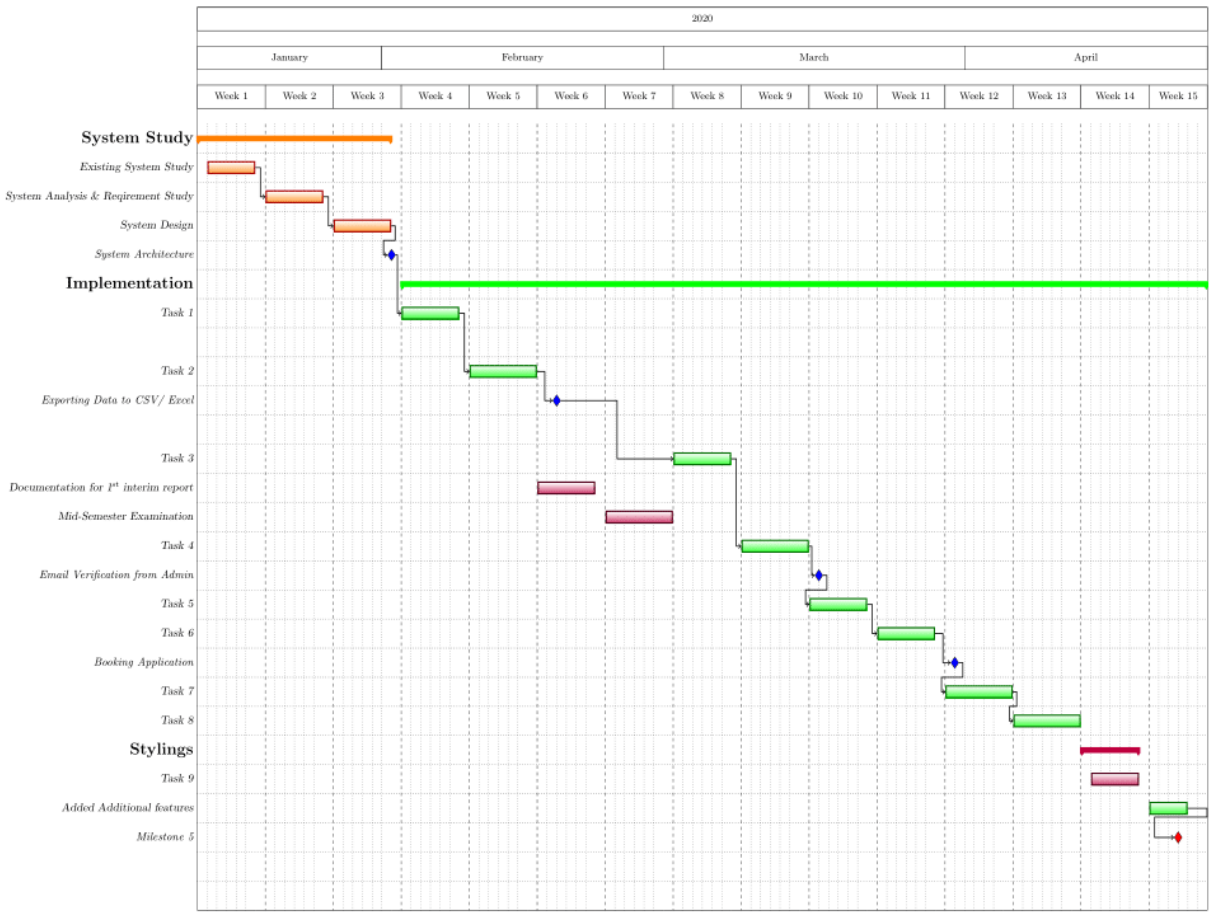


Figure 1.2: Progress Report

The various tasks are described inline below:

- **Task 1:**

    1. Created Admin Page using Django
    2. Populate the database by taking 4-5 samples of data
    3. Created forms and customized Admin

[1] as submitted to the supervisor dated 30 April 2020

- **Task 2:**

    1. Build the Template
    2. Made interface to populate the database through frontend
    3. Displayed the data through frontend

- **Task 3:**

    1. Added Search link with filters
    2. Added options for Various operations on frontend for Admin as well as Real users
    3. Added Export (to CSV/ Excel) link (Milestone Achieved)

- **Task 4:**

    1. Added Choic fields
    2. Established link of fields with Database table
    3. Validated the form on both frontend as well as backend
    4. Added checks to prevent duplicates

- **Task 5:**

    1. Created login/ Sign-Up page
    2. Added Email Verfication(Milestone achieved)
    3. Validated the invalid Email addresses
    4. Provided Pop-Up notification messages

- **Task 6:**

    1. Explored more on User Authentication & implemented
    2. Implemented User permissions with verification
    3. Added setting section to the Application

- **Task 7:**

    - Added Booking Application with the following features (Milestone):
        1. Reserve the properties
        2. Booking Status
        3. Booking errors
        4. Booked List
        5. Extra person-infos

- **Task 8:**

    1. Added and Customized the Registration profiles
    2. Generated a Unique-ID alongwith their activation key for each registered User

- **Task 9:**
    1. Added Static files
    2. Added Bootstrap
    3. Added jquery
    4. Crispy forms
- **Additional features:**
    1. Explored Django-callers
    2. Implemented callers
    3. Explored Speech Recognition using django
- **Current Milestones:**[2]
    - Callers related errors to be debugged
    - Speech Recognition facility to be featured in the application

The milestones were completed[3] successfully with fixing all the bugs and the final touches were given to the frontend.

[2]as on 30 April 2020
[3]Completed on 15 May 2020

# Chapter 2

# Related work

## 2.1 Existing System Study

Three websites have been compared those of which are namely *Cloudbeds*[1], *eZeeFrontdesk* [2] *and Frontdesk Anywhere*[3] as a part of my Existing System Study.While comparing all the three websites I found that all have the basic features which are common in almost all the property management systems.Basic features as booking,check-in,check-out details etc.None of them has authentication and autorization features for various layered users.Some of them are very particular to only hotels or room on rents, none of them are having the scene of multiple layers of properties.None of them had maintained a proper database at the backend that can be accessed from the frontend.None of the single website has modern features such as E-mail Verfication,Caller Speech Recognition etc. They still lag behind from security point of vies as well as modern features.

## 2.2 Comparative analysis

No doubt all the sites are well maintained with fascinated stylings and formats, but the issue is they when we search for the best sites for property management system, they all are relevant to the hotels, and its notifiable that they have only focused on the *Hotels* as their only assets or to some extent the restaurants. We know that the properties in this world is not limited to only hotels and restaurants, as their requirements are necessary but not in other fields like *Institutes*. This project has dealt with the particular institute ***IIIT Senapati*** that itself comprises various departments incorporating numerous classrooms and the laboratories which are essential as a part of the education. The explored system narrates their own story which are limited to only hotels and not the nearby resorts and restaurants and the tourists points.
The fundamental basic requirement necessary for building up the system which I found missing while analysing and doing the comparative assessments is that they lacks with the three layered architecture, i.e., from underlying concepts to the final view to the real world users. The consistency has not been maintained among the data flow.

[1]https://www.cloudbeds.com/ (15 January 2020)
[2]https://www.ezeefrontdesk.com/ (16 January 2020)
[3]https://www.frontdeskanywhere.com/ (18 January 2020 )

## 2.3 Method(s) used for existing system study

The three important methods under qualitative analysis is used for studying existing system which are as follows:

1. Observation: This includes collecting data as observed by an observer from the frontend such as services provided, features, security related issues,how far thier services can be stretched etc.This method is sometimes known as Participant Observation and are often termed as appropriate for collecting data on naturally occurring behaviors in their usual contexts.

2. In-depth interviews: These are optimal for collecting data on individuals personal histories, perspectives, and experiences, particularly when sensitive topics are being explored.These includes rating as given by users who already used once and experienced with their performances.This can be consider as a very important tool for comparative qualitative analysis.

3. Issues and questioning:This method helps to get information about the issues that have been faced or still facing with the existing systems. This can be used to modify the existing system as well as coming up with new solutions and exciting features.

## 2.4 Summary

Some of the security issues still persist in the existing system. Does not have well maintained database as well as frontend for three different layered users i.e., Superuser,Managers and Customers.Authentication and Authorization features are still missing. The new features and operations that the proposed project is going to have is that the first and the foremost of having multiple layers not paticular with only one type of property such as only hotels, giving pdfs of the bill so generated, E-mail Verification, Caller giving details of the system and how to use it, and most interesting feature of speech Recognition which can be proved to be the most user-friendly in this management system. Keeping all the basic features well maintained such as booking,searching,registering this project is going to have a proper database which can be accessed from the frontend having authorization constraints.

# Chapter 3

# System Design

## 3.1 System design

The system is so designed and planned to have the admin site which can only be accessed by the authorized superuser. The superuser can also view the frontend site from the backend as and when required. The managers can access the backend database only after authorized to do so. A User/ Customer can register themselves as a new user which then can get E-mail verfied followed by having details as data in the database from where they can receive their acivation key which can further be used for various purposes such as booking. Users can search different properties from catalog as upon their wish and book them as needed. They can be able to view full details of the property of their choice once they have been logged in as user as well as they can have their own excel sheet for the properties in the catalog from where they can refer in future.

## 3.2 Architecture

### 3.2.1 The Three Schema Architecture

My management system follows the classic Three layered Architecture, one of the principles of ***Database Management System***(DBMS). Referenced books are [1], [2].

1. **External Level**: At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database. Compared to the designed management system, this level constitutes only the end users, who have their own perspectives to use relevant database contents , and can only allowed to view it and rest of the data are hidden from them. More formally, Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.

2. **Conceptual level**: The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level. This also describes the structure of the whole database. Comparing with my management system , this level comprises the admin staff who looks after various tasks of DDL, DML,DCL so as to maintain the system properly.

3. **Physical Level**: The internal schema is also known as a physical schema. It uses the physical data model. It is used to define that how the data will be stored in a block. This lowest level, also known as internal level includes all the superuser who actually decides the overview of the system and how to store the data into the database.

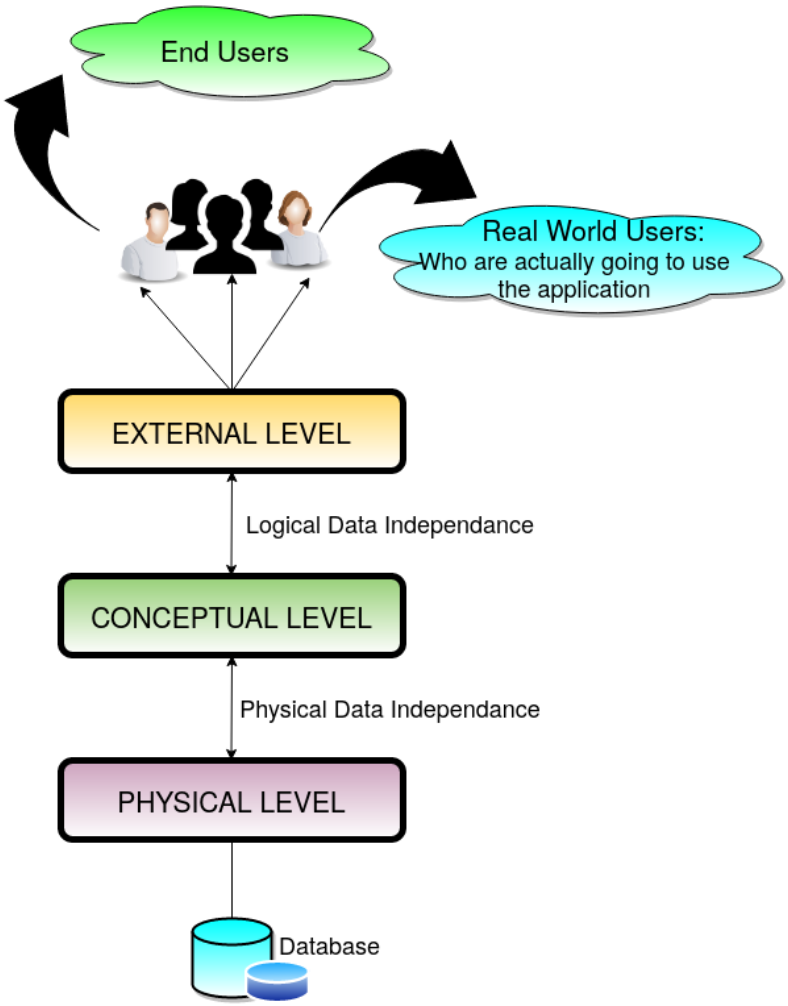


Figure 3.1: Three Schema Architecture

The three schema architecture is also called *ANSI/SPARC architecture* or three-level architecture. This framework is used to describe the structure of a specific database system.
***Logical Data Independence*** is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk. Logical data independence is used to separate the external level from the conceptual view and it occurs at occurs at the user interface level.
***Physical Data Independence*** Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema. If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected. Physical data independence is used to separate conceptual levels from the internal levels. and occurs at the logical interface level. Referenced books are [3],

## 3.3 System requirement Analysis

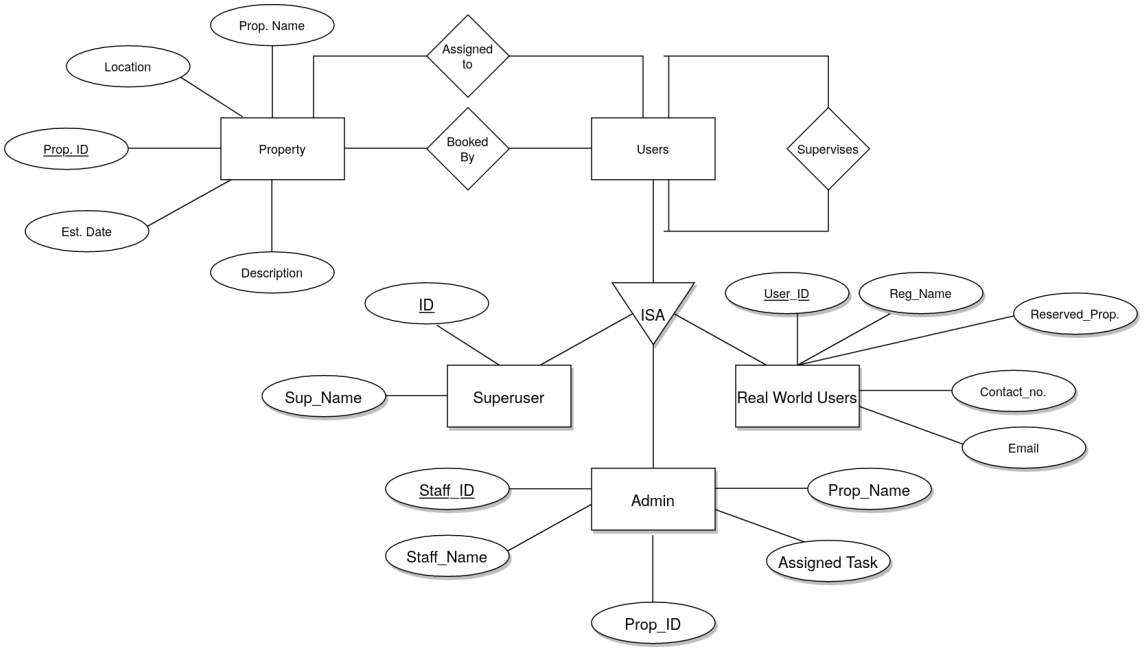


Figure 3.2: The ER Model

The basic purpose of the conceptual ER model is then to establish structural metadata commonality for the master data entities between the set of logical ER models. The conceptual data model is used to form commonality relationships between ER models as a basis for data model integration. Referenced books are [4], [5]. The above ER model is designed with two fundamental entities:

1. **Properties:** The properties or the assets are one of the basic entities on which the system is being built upon. The various attributes are so chosen to get the clear recognizable property of the user's desired as and when required. The *Prop.ID* will uniquely identify the specific property or the asset from the database along with the complete details that includes its establishment date and the managing staff for the particular property. The property entity has been framed with the following attributes:

   - Prop. ID
   - Property Name
   - Location
   - Establishment Date
   - Description

2. **Users:** As it is clear from above that the users entity supports ***ISA Relationship types***. An ISA relationship suggests that when an entity type contains certain entities thathave special properties not shared by all entities, this suggests two entity types should be created with an ISA relationship type between them. Here, with reference to this system under the project, we can say that the user entity is in ISA relationship with the ***Superuser, Admin Staffs and Real World Users***. All the three party users of the system have their own

charaterized features and attributes, but one thing that is common in all is the *ID*. This ID uniquely defines each one of them, subsequently helpful for differentiating among themselves. For instance, let's take the scenario of a member of the system and we want to know whether it's an admin staff or the maintainer, in such a case the specific ID provided will help us to know their identity.

(a) **Superuser:** The superusers group includes the higher-level authorities who investigates the whole system to check the functionalities and keep a track of the regular operations performed by the admin staff members. This entity only requires the two attributes:
   i. ID
   ii. Name

(b) **Admin:** The Admin staffs are those members who are responsible for various DBMS operations such as *Data Manipulation Language(DML), Data Definition Language(DDL), Data Control Language(DCL)* and many more. These colleagues are the real maintainers and developers to maintain the actual flow sequence. They acquires the following:
   i. Staff-ID
   ii. Staff-Name
   iii. Assigned task
   iv. Assigned Property
   v. Property ID undertaken

(c) **Real World Users:** These comprises the actual users who are going to use the frontend of the sytem and can give feedbacks and comments on the existing systems. They have the following attributes:
   i. User-ID
   ii. Registration-Name
   iii. Reserved Property
   iv. Contact Details
   v. Email Address

Apart from all these, the designed system has the following pre-defined relations:

- ***Assigned-to:*** This relation implies that the various properties have been assigned to the number of admin staffs in order to have better functionality. Each of the admin staff members have been assigned a particular asset or the property whose sole responsibilities are relevant to that specific property. With these responsibilities they can be able to entertain the users in a more better and improved manner with respect to any enquiry and the confirmation related details.

- ***Booked-by:*** The users can book and reserve the properties as and when required on their own self desires. The *booked-by* relation is the most perfect matched relation with the properties and the real world users.

- ***Supervises:*** This is one of the recursive relations involved in the ER model. This indicates that the superuser keeps an eye on the overall performance of the system and suggest some changes to be done in order to minimize and eradicate the hurdles and issues confronted by any of its staff members or the real world users. It is a kind of a unary relationship, also called recursive, is one in which a relationship exists between occurrences of the same entity set i.e., superuser and the admin staffs.

# Chapter 4

# Implementation

## 4.1 Django: An overview

Django, a web framework is being used to implement the propsed project which follows model-template-view architectural paradigm. Various packages are present that helps to create the application user-freiendly such as Registration,Reservation and a number of various others.Django models are used as a structure to define fields and their types which will be saved in the database. Whatever changes we want to make in the database and want to store them in the database permanently are done using Django Models.According to Django Models, A model is the single, definitive source of information about the data. It contains the essential fields and behaviors of the data which are storing. Generally, each model maps to a single database table.

1. Each model is a Python class that subclasses django.db.models.Model.
2. Each attribute of the model represents a database field.
3. With all of this, Django gives us an automatically-generated database-access API.

Django basically uses a bottom-up approach as it starts from creating small models and adding them up together.The basic advantage is that redundancy is eliminated.Data types used are similar to the one used for basic SQL Database. Step-by-step verification , authentication and authorization are taken careof at each step. Django Admin interface reads the metadata of the models we are created,to provide a quick, model-centric interface where trusted users can manage content on the site. The admins recommended use is limited to an organizations internal management tool. Its not intended for building your entire front end around.The basic workflow of Djangos admin is, in a nutshell, select an object, then change it. This works well for a majority of use cases.
Booking Model is so implemented as to provide easy to use,smooth transactions and quick access to prevent the information.The project is also going to foucus on creating a sense of urgency:adding elements that emphasize the best deals and showing real-time dynamics helps users find the best propositions.Finally we will make the Decision-making process for the users quite simple to visit only the created site.

## 4.2 Software Model Design

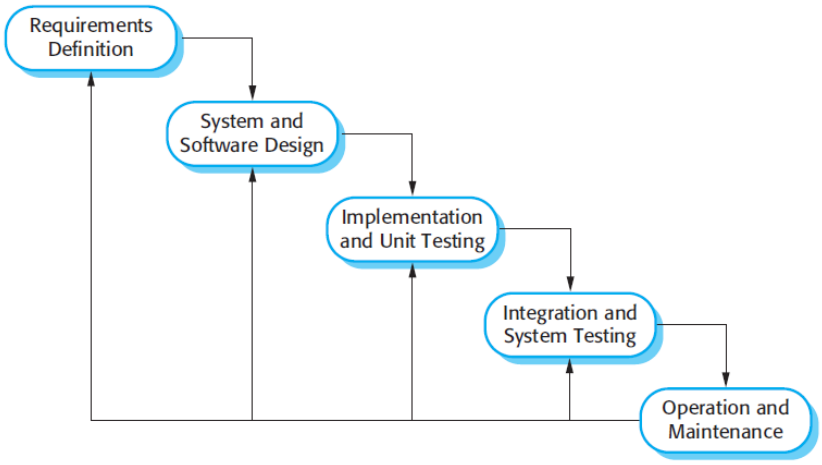


Figure 4.1: The classis Software Modelling

Software design is the process of defining software methods, functions, objects, and the overall structure and interaction of the code so that the resulting functionality will satisfy the users requirements. The above model is well known as the ***Waterfall Model*** . As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a Linear-Sequential Life Cycle Model. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

The first two stages have been described in ***System Requirement Ananlysis***, in which I have described the ER model describing the various needed entities along with their necessary attributes. In the subsequent sections, I will describe the system flow sequence with the help of data flow diagrams. The implementation portion will get more understanding with the UML diagrams and object modelling.

## 4.3 Experimental set-up description

### 4.3.1 Flow Sequence

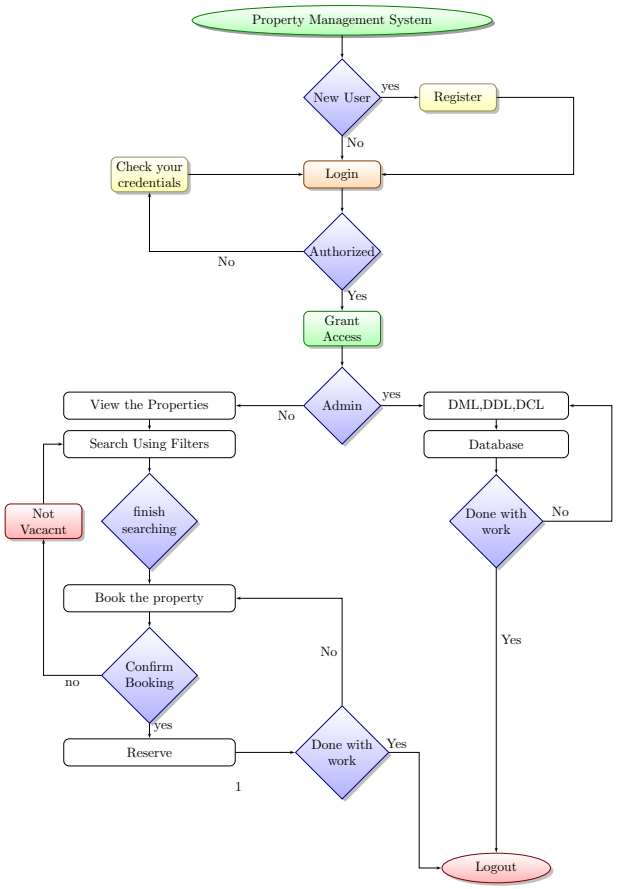


Figure 4.2: Data Flow Diagram

With refernce to the mentioned Figure 4.1, it is clear that the system starts operating from its initial stage of authenticating the user from entering into the system whether it is an admin staff, or the superuser, or the real world user. Each one of the participating members must login and after successful completion of the work they should be to logout.
The system is so designed as to make it operable sequentially initiating from login mechanism to fulfill the requirements For authentication. After successfully verifying the authorization the system will open up a new window separately for both, the admin as well as the real world users. The admin will look after DML,DDL,DCL while the users will engage with their own activities of searching and booking. Having done with works, they can logout. the system is so designed such that all the different layered members will perform their tasks and operations effectively and efficiently.

### 4.3.2 Class and Module Description

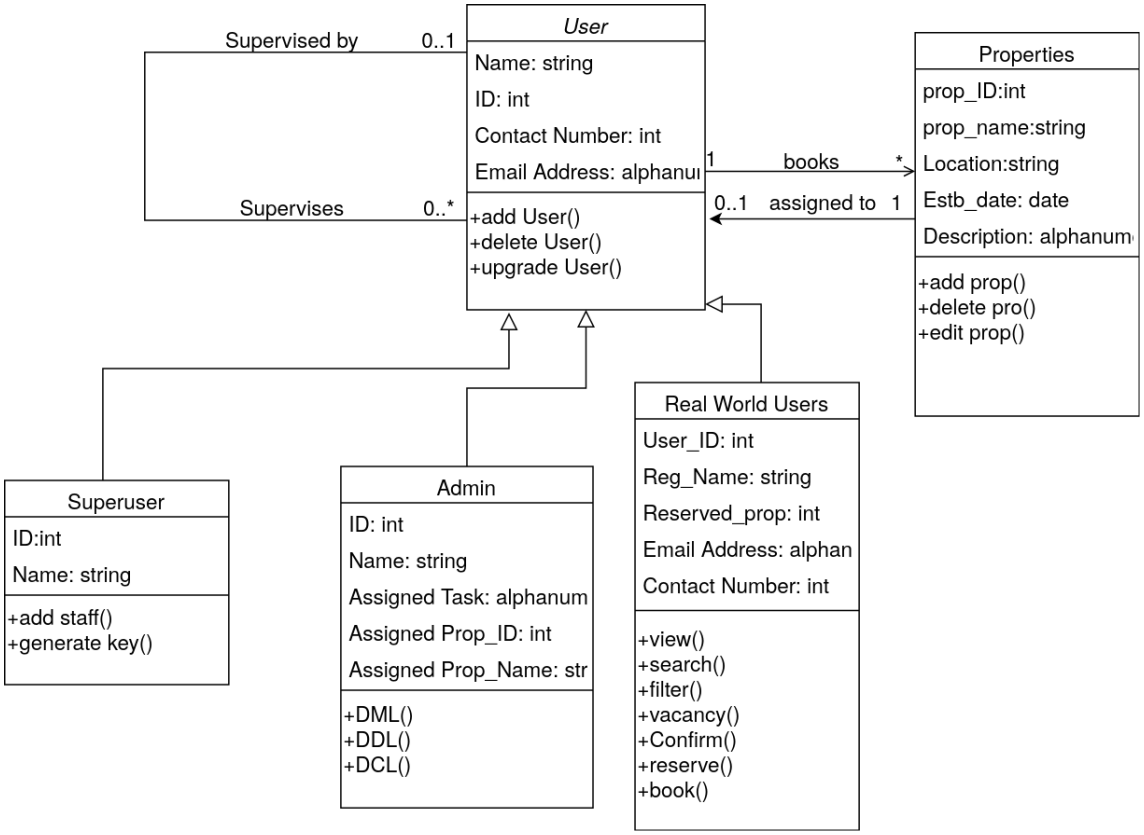


Figure 4.3: Class Diagram

The above class diagram demonstrates the various classes and modules aming them of which comprises the users and properties. This is one of the modified versions of the previously discussed ER model. It described the the used data types for various attributes and the incorporated functins associated with each of them. The various functions associated with the users are ***View, Search, Filter, Confirmation and Reservation***. It is clear from above that the admin staff is responsible for various languages associated with DBMS. The fundamental functions associated with properties are:

- Add Properties
- Delete Properties
- Edit the Prperties

These are the basic functionalities supported by the system.

### 4.3.3 Users Perspective: The Front View

The various aspects an users expects from the system and which is my one of the main ideal outcomes of the project can be viewed from the following figure:

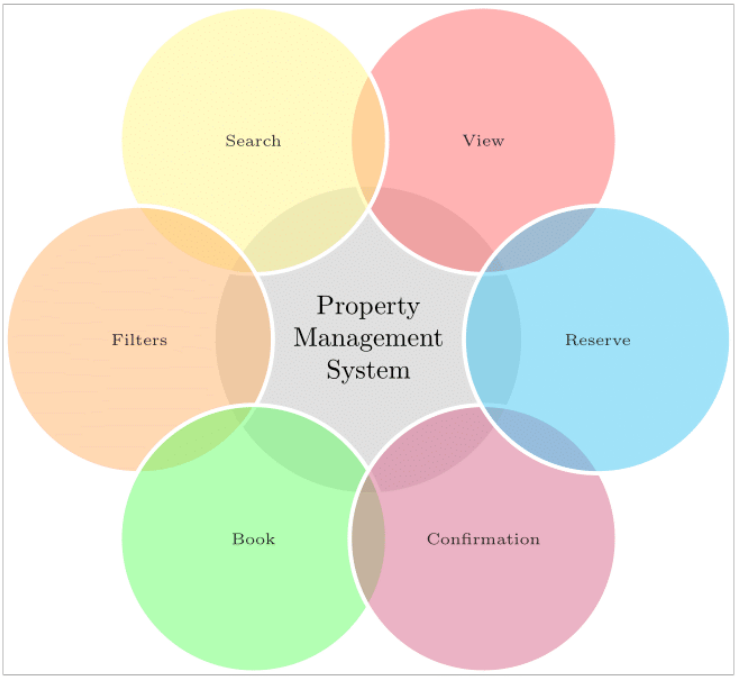


Figure 4.4: Operations offered to the users

The above figure portrays the flexibilities and accessibilities provided to the users in order to perform various operations on the properties included in the databases. These are:

1. ***View:*** The users can be able to properly view the property database along with its full description in order to perform the next task of searching of the desired assets.

2. ***Search:*** This facility will enable the users to search the property from a huge list of properties as contained in the database.

3. ***Filter:*** This flexibility will help the user to filter the search by feeding up some necessary details such as location and name.

4. ***Book:*** The accessibility to various assets and prperties can be done through the booking slots and the vacancy information.

5. ***Confirm:*** Before taking the final step of reserving the property, it is mandatory to check any conflicting reservation by other parties and hence to resolve it.

6. ***Reserve:*** This the final step taken by the user to reserve the property according to their necessaties and requirements.

## 4.4 Obtained result

The generated output can be viewed from the screen captures included in this document:

### 4.4.1 Outlook

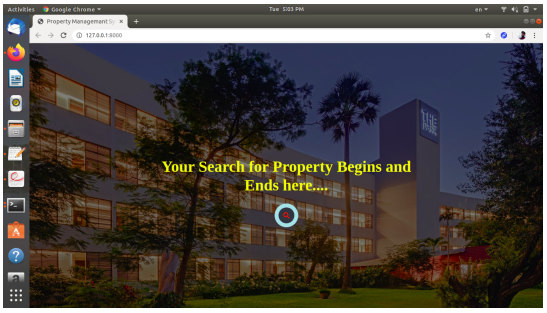


Figure 4.5: Overlook of the system

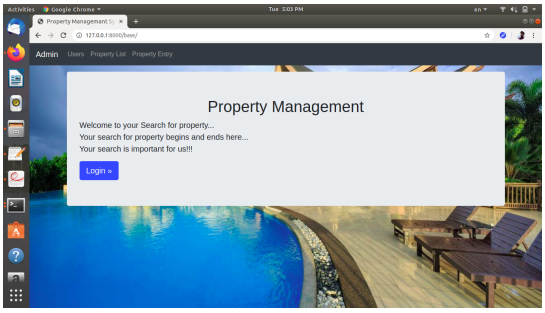


Figure 4.6: Login Page

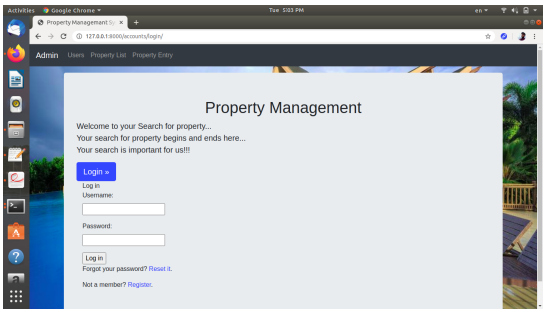


Figure 4.7: The Email Verification Page

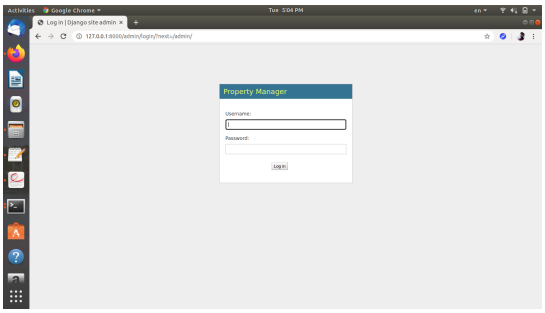


Figure 4.8: The Admin Login Page

### 4.4.2 The Backends

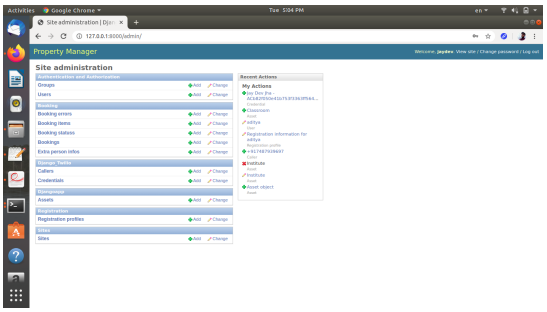


Figure 4.9: The Site Administration

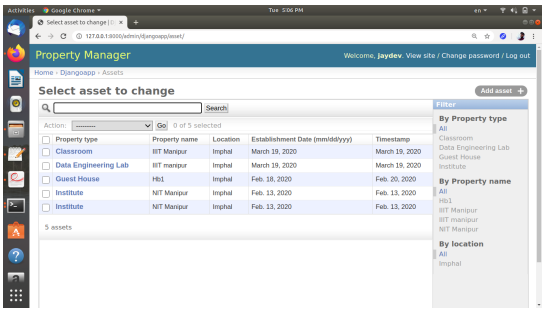


Figure 4.10: Overview of the database

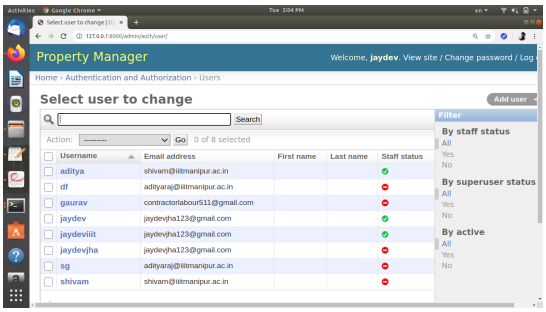


Figure 4.11: Superuser's authority to change the staff status

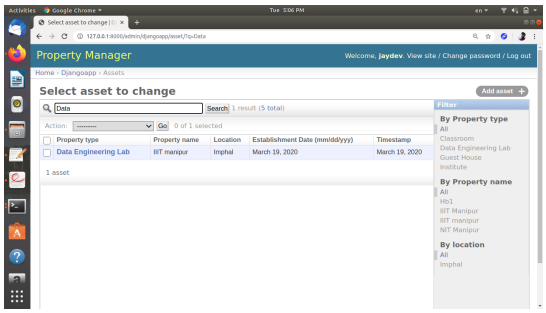


Figure 4.12: Filter Search

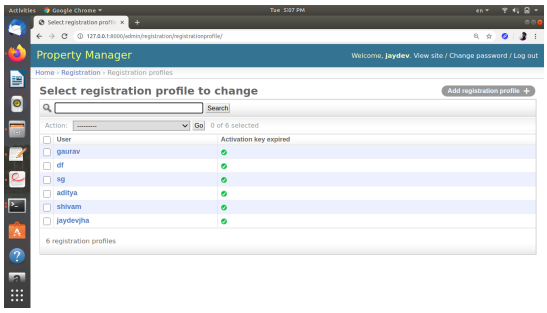


Figure 4.13: Registration profiles for the users

### 4.4.3 Booking Facility

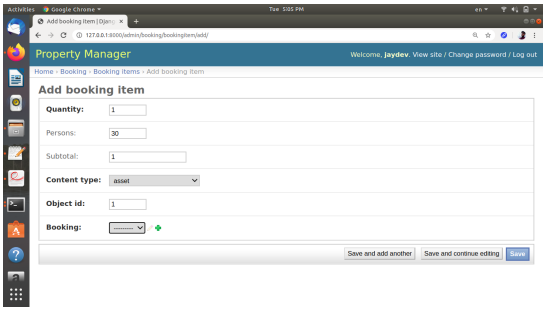


Figure 4.14: Overview of booking facility

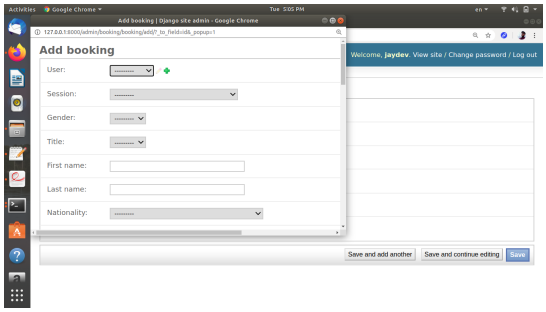


Figure 4.15: Booking Form-I

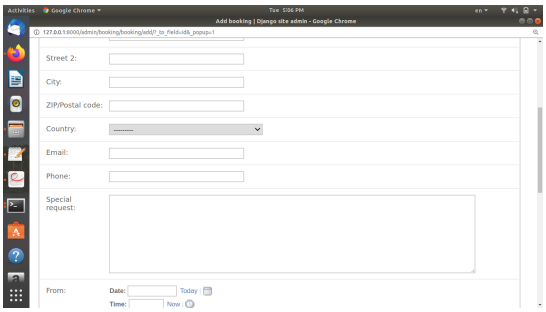


Figure 4.16: Booking Form-II

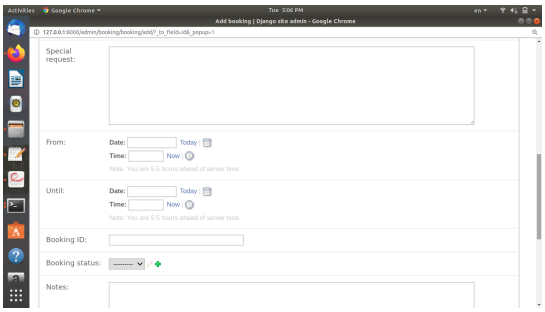


Figure 4.17: Booking Form-III

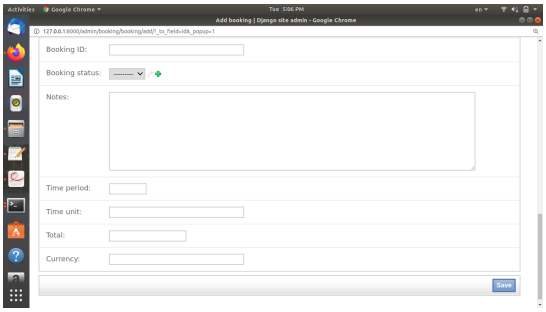


Figure 4.18: Booking Form-IV

### 4.4.4 Front View

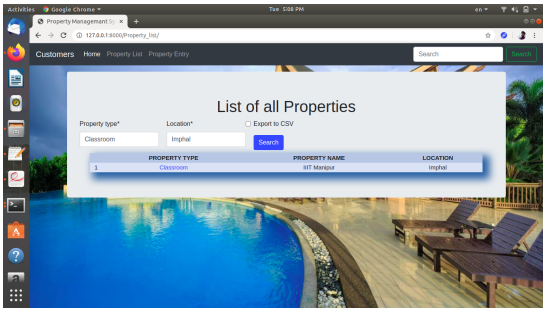


Figure 4.19: Filter Search

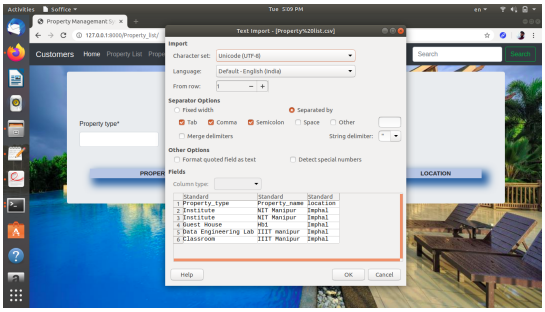


Figure 4.20: Export to CSV

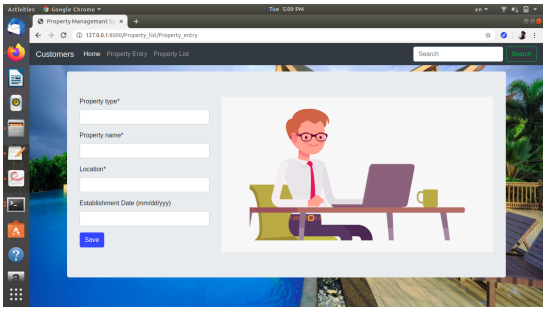


Figure 4.21: Add Properties from frontend

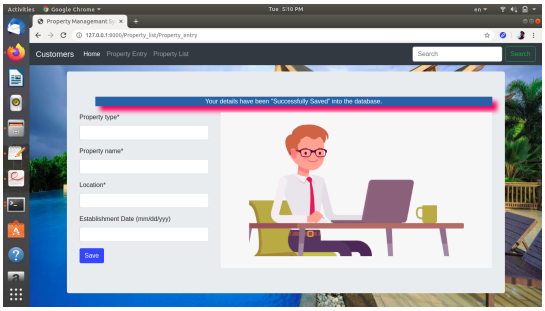


Figure 4.22: Pop message for any operation

# Chapter 5

# Result analysis and Testing

The results and the outcomes have been generated as expected and desired. The project has been designed and goes perfectly well from all the three prespectives i.e., ***The overview or the outlook, The Backend portion and the Front View***, all have proven to support with the design scheme and fulfill the system architecture. The data is being flowing into the system via the three hierarchal layers that is from the superuser to final the viewer consitutuing the real world user going sequentiallythrough the administration staffs.

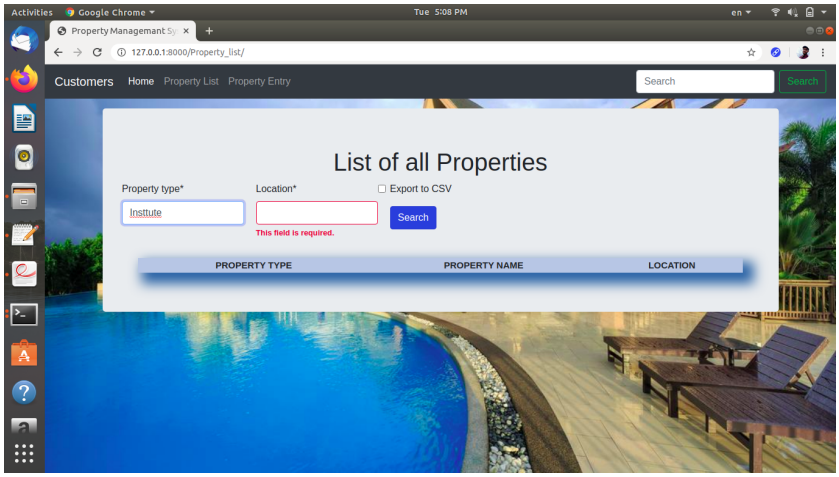


Figure 5.1: A simple test Case

Various use cases has been tested as one shown above. The above screeshot clearly suggests there must the be location, i.e., the location details is mandatory to fetch the desired and available property from the database. The location must be feeded up in the search box, otherwise it will pop the warning message that ***this field is required***. Other test cases include the ***Email verification*** whether it is reacheable to the user signing up. It also checks whether the email is valid.

Apart from frontend checks, uses cases were also tested on the backend portion, that whether the superuser is able to keep a log of its admin staffs, whether the keys are being generated for particular user and staff members. It has also checked that whether the information are being retrieved properly and all the queries are functioning properly. The system has also been tested whether there is none of the broken links between frontend as well as the actual front site.

# Chapter 6

# Conclusion and Future work

## 6.1 Conclusion

On Successfully completing the project, the Property management System is able to deliver its flexibility, compatibility and accessibility to all its users. Apart from this the system has been able to:

- Achieve the ideal objective of enhancing the filter search among properties
- Deliver the booking functionality with vacancy details
- Maintain the data flow sequence and support the three schema architecture

The project has been successfully completed giving the expected outcomes as planned before. The ***Property Management System*** is able to perform and support various of its functionalities as designed in system design section. It can be conclude that the system has been designed keeping in mind the specifications of the system. Overall the project also teaches the essential skills like:

- Using system analysis and design techniques like data flow diagram in designing the system.
- Understanding the database handling and query processing.

To put it in a nutshell ***Working on the project was good experience. I understand the importance of Planning and Designing as a part of software development.***

## 6.2 Future Scope

The future scope of this project includes the following:

1. Extending the database
2. Adding google routes facility to approach
3. Speech recognition facility
4. Extending the system to be used for a city
5. Collaborative forums to be used the propsed in a state

Nothing is perfect in this world. So, we are also no exception. Although, we have tried our best to present the information effectively, yet, there can be further enhancement in the Application.

We have taken care of all the critical aspects, which need to take care of during the development of the Project.

Like the things this project also has some limitations and can further be enhances by someone, because there are certain drawbacks that do not permit the system to be fully accurate.

The system is highly flexible one and is well efficient to make easy interactions with the client. The key focus is given on data security, as the project is online and will be transferred in network. The speed and accuracy will be maintained in a proper way.

This will be a user-friendly one and can successfully overcome strict and severe validation checks. The system will be a flexible one and changes whenever can be made easy. Using the facility and flexibility in .NET and SQL, the software can be developed in a neat and simple manner there by reducing the operators work.

# Bibliography


[1] S. R.Elmasri, *Database Systems: Models,Languages,Design and Application Programming.* 6 Edition: Pearson Education India, 2013.

[2] A. Silberschatz, *Database Systems Concepts.* 6 Edition: Tata McGraw Hill, 2019.

[3] O. M. T, *Principles Of Distributed Database Systems.* 3 Edition: Springer Exclusive(Cbs), 2012.

[4] C. J. Date, *An Introduction to database systems.* 8 Edition: Pearson, 2003.

[5] C. B. T. connolly, *Database Systems: Apractical Approach to Design,Implementation and Management.* 6 Edition: Pearson, 2014.